

REMARKS

I. Introduction

Claims 1-31, and 33-42 are pending in the present application. Claim 32 is canceled in this Response. Claims 38-42 are added in this Response. In a June 30, 2004 Office Action (herein "Office Action"), Claim 32 was rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 1-9, 12-21, and 24-37 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,282,454 to Papadopoulos et al. (herein "Papadopoulos"). Claims 10, 11, 22, and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Papadopoulos in view of U.S. Patent No. 6,453,687 to Sharood et al. (herein "Sharood").

The amendments made to Claims 1, 2, 4-6, 15, and 28 are presented for clarification purposes only and do not modify or limit the scope of the claims in any way. In particular, independent Claims 1 and 27 each include the limitation of a "non-markup language Web site database." Subsequent references that originally only recited "said Web site database" have been amended to clarify that "said Web site database" refers to the limitation of a "non-markup language Web site database." The amendment made to Claim 35 is to correct the dependency to properly depend from independent Claim 27.

II. Claim Rejections

A. 35 U.S.C. § 112, Second Paragraph

Claim 32 was rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that applicants regard as the invention. Claim 32 has been canceled.

B. 35 U.S.C. § 102(e)

1) Introduction

Claims 1-9, 12-21, and 24-37 were rejected under 35 U.S.C. § 102(e) as being anticipated by Papadopoulos. For the following reasons, applicants respectfully submit that the rejected claims of the present application are not anticipated by Papadopoulos, because Papadopoulos fails to teach or suggest a system or method for providing information regarding the operation of a control system that receives non-markup language configuration data defining a Web site that is stored as a non-markup language Web site database (Claims 1-18, and Claims 27-31, and 33-37). Likewise, Papadopoulos also fails to teach or suggest an apparatus for providing information regarding the operation of a control system that is operative to store a non-markup language Web site database defining a Web site associated with the control system, wherein the non-markup language Web site database is in a format that may be utilized by a central processing unit to dynamically render Web pages of said Web site, as called for in Claims 19-26. Prior to discussing more detailed reasons why applicants believe that all of the pending claims of the present application, as amended, are allowable over Papadopoulos, a brief description of the present invention and the cited references is presented.

(a) Summary of the Present Invention

The present application generally relates to the field of process control systems. More particularly, the present application is directed toward providing a method, system, and apparatus for providing information regarding the operation of a control system, such as a Programmable Logic Controller ("PLC") that does not require a user to create a Web site using a markup language (i.e., the user may create a Web site using a non-markup language). Additionally, upon receipt of a non-markup language that defines a Web site, the claimed invention stores that data in a manner that requires less memory space than storing conventional markup language Web pages.

In an embodiment, the invention includes a Web server module associated with a control system that provides a Web site defined utilizing a remote computer system and a Web server module configuration application. The Web server module configuration application provides an easy-to-use interface for defining the Web site provided by the Web server module. *See Application*, p. 12. In particular, the Web server module configuration application allows a user to define the Web site using a non-markup language by providing the user with easy-to-use menus and interfaces. The menus and interfaces allow a user to design a web site through selections on a number of display screens, rather than through the use a markup language. *See id.* The Web server module configuration application stores the user-made selections in a non-markup language Web site database and transmits that database to the Web server module. *See id.* The Web server module utilizes the contents of the non-markup language Web site database to dynamically generate markup language pages for a user when a request is received. *See id.*

FIGURES 28-34 illustrate screen shots of a number of display screens that are provided to a user by the Web server module configuration application that allows the user to design a web site without requiring a user to understand a markup language or programming. For example, referring to the screen illustrated in FIGURE 28, the Web server module configuration application allows a user to configure the Web server module by providing a module reference 254, a rack designator 256, a slot designator 258, and a host name 260. *See id.* at 32. Additionally, as illustrated by FIGURE 29, a user may easily add Web pages, subtract Web pages, or modify pages of a Web site provided by the Web server module. *See id.* at 33. Still further, as illustrated by FIGURES 30 and 31, the Web server module configuration application allows a user to create tables and to specify the tagged data items and columns that are to appear in the tables. *See id.* at 33. As further described in the specification of the application, the Web

server configuration application allows a user to completely define the operational characteristics of a Web server module and the Web site to be provided by that module. *See id.* at 32-34.

Once a user has defined the Web site, via the Web server configuration application, the Web server configuration application stores the information as a non-markup language Web site database, also referred to in the application as a screen database (e.g., FIGURE 5, element 112) and transmits that database to the Web server module. *See id.* at 15. The non-markup language database fully defines the Web site to be served by the Web server module. *See id.* When requested, the Web server module utilizes the contents of the non-markup language Web site database to dynamically generate markup language Web pages. *See id.* Storing the non-markup language Web site allows a user to create a database which does not just store Web pages, but instead stores all attributes required to dynamically generate pages of the Web site when requested.

Numerous advantages may be realized by the system, method, and apparatus recited in the claims for the present application. In one aspect, a user may define, from a remote location, a Web site for providing information regarding the operation of a control system without having to use a markup language. As discussed in the Background section of the application, and as discussed below with respect to Papadopoulos, the main drawback with existing web server based systems for accessing data contained in the memory of a control system is the difficulty in creating and modifying the Web site that is provided by the Web server module, as well as associating the PLC data with table entries or non-text renderings within the markup language format. This process is typically an arduous one that involves an operator creating each of the Web pages of the Web site using a standard markup language, such as the hyper-text markup language ("HTML") or extensible markup language ("XML"), and possibly a programming language such as JAVA®. While PLC operators are often well versed in ladder logic, HTML, XML, and JAVA are typically foreign topics. Providing a system, method, and apparatus that

allows a user to define a Web site using non-markup languages resolves this drawback of existing server based systems such as Papadopoulos.

In another aspect, embodiments of the present invention store data defining a Web site in a manner that requires less memory than storing conventional markup language Web pages. Web servers generally use non-volatile memory to store conventional markup language Web pages and any associated information, like graphics. Typically, a standard file system is created within the non-volatile memory, with all the HTML contents for a page rendering stored there. For large Web sites, such a storage scheme results in a large amount of non-volatile memory being consumed. The claimed invention, in contrast, only stores the information necessary to generate a Web site when requested, thereby reducing the memory requirements.

(b) U.S. Patent No. 6,282,454 to Papadopoulos et al.

Papadopoulos is purportedly directed toward a web interface to a programmable controller. The stated goal of Papadopoulos is "to develop an automation control system whereby an user could use general, commercial networks such as the Internet in place of specialized industrial networks to remotely monitor automation control devices such as PLCs." Papadopoulos, Col. 2, ll. 25-30. In accordance with that goal, Papadopoulos describes "a system of essential elements including...a Web interface, a local network and a network interface to at least one PLC control system....The Web interface runs Web pages from an Ethernet board coupled directly to the PLC back plane and includes an HTTP protocol interpreter, a PLC back plane driver, a TCP/IP stack, and an Ethernet board kernel." Col. 2, ll. 45-54.

In using the system, Papadopoulos describes that a user can enter the address of the Web site which will display a home page that may contain text, some type of multimedia offerings such as graphic images, video, or audio, and possibly hypertext links to other documents. The Web site home page, and other pages of the web site are provided by the web server. Col. 4, ll. 10-15. As further described in Papadopoulos:

The home page contains hyperlinks to seven pages of data. The configuration page will display the configuration of PLC 32. The remote I/O and distributed I/O module health status pages are a series of linked pages. The first page displays the communication health statistics at the Remote I/O and Distributed I/O head and contains a link to a configured drop page. The configured drop page displays a table containing drop numbers which are linked to a drop status page and rack numbers which are linked to the drop and rack configuration pages. Two tables are included in the drop status page, one for showing the communication status of the drop and the other for showing which racks are populated with the I/O modules. The drop and rack configuration page displays the I/O modules, their health, and slot location for the given rack. From a selected module, a user can view its input and output values. Register data is displayed in a template having a form and a table, with the user entering an address and a length. The table will display the registers values. A table showing option modules and their slot location is displayed on the back plane configuration page. The data appearing on the pages is static but can be automatically updated at preselected times.

Id. at Col. 8, l. 62 - Col. 9, l. 16.

While Papadopoulos describes the ability for a user to obtain snapshots of the status of a control system from a remote location via seven predefined web pages, Papadopoulos fails to teach or suggest a system, or method for providing information regarding the operation of a control system that receives non-markup language configuration data defining a Web site that is stored as a non-markup language Web site database (Claims 1-18, and Claims 27-37). Likewise, Papadopoulos also fails to teach or suggest an apparatus for providing information regarding the operation of a control system that is operative to store a non-markup language Web site database defining a Web site associated with the control system, wherein the non-markup language Web site database is in a format that may be utilized by a central processing unit to dynamically render Web pages of said Web site. Papadopoulos is limited to a system that uses conventional markup language web pages to provide snapshots of control system information.

2) The Claims Distinguished

(a) Claims 1, 19, and 27

Claim 1 reads as follows:

A system for providing information regarding the operation of a control system, comprising:

a Web server module associated with said control system, said Web server module having a memory operative to store a non-markup language Web site database defining a Web site; and

a computer operative to receive non-markup language configuration data defining said Web site, to store said configuration data as said non-markup language Web site database, and to transmit said non-markup language Web site database to said Web server module.

Claim 19, as amended, reads as follows:

An apparatus for providing information regarding the operation of a control system, comprising:

a central processing unit;

a memory coupled to said central processing unit operative to store a non-markup language Web site database defining a Web site associated with said control system, said non-markup language Web site database in a format that may be utilized by said central processing unit to dynamically render Web pages of said Web site;

a first interface coupled to said central processing unit for communicating with said control system controller and utilized by said central processing unit to retrieve and provide information regarding the operating of and for said control system;

a second interface for communicating with a remote computer also coupled to said central processing unit and utilized by said central processing unit to receive requests for said Web pages and to transmit responses to said requests.

Claim 27 reads as follows:

A method for providing information regarding the operation of a control system, comprising:

receiving non-markup language configuration data defining a Web site;

storing said configuration data as a non-markup language Web site database; and

in response to a request, dynamically generating a Web page defined by the non-markup language configuration data stored as a non-markup language

Web site database that provides information regarding the operation of a control system.

As recited above, Claims 1, 19, and 27 describe a system, method, and apparatus for providing information regarding the operation of a control system. Both Claims 1 and 27 include the limitations of receiving non-markup language configuration data defining a Web site and storing that data as a non-markup language Web site database. Additionally, Claim 27 also includes the limitation of "dynamically generating a Web page defined by the non-markup language configuration data stored as a non-markup language Web site database that provides information regarding the operation of a control system." Similarly, Claim 19 includes the limitation of "a non-markup language Web site database defining a Web site associated with a control system, said non-markup language Web site database in a format that may be utilized by said central processing unit to dynamically render Web pages of said Web site."

Providing a system, method, and apparatus that utilizes a non-markup language to define a Web site, stores that non-markup language instead of conventional markup language web pages, and dynamically generates web pages based on that information, provides the ability for a user to easily create a web site without having to know a markup language. Additionally, storing information that may be used to dynamically generate a web page, as opposed to storing conventional web pages and associated information, reduces the total amount of required memory.

The Office Action asserts that Papadopoulos teaches: "a Web server module (fig. 3, element 30)...having a memory operative to store a non-markup language Web site database defining a Web site (fig. 3, element 52);" "a Web site database defining a Web site associated with said control system, said Web site database in a format that may be utilized by said central processing unit to dynamically render Web pages of said Web site (F3, E52);" "a control system, comprising: receiving non-markup language configuration data defining a Web site (F1, E20, C3, L48-60);" and "storing said configuration data as a non-markup language Web site database

(F2, E30, C4, L9-12)." Office Action, pp. 3, 7, and 9. For the following reasons, applicants respectfully assert that Papadopoulos fails to teach each of the limitations of Claims 1, 19, and 27.

In contrast to Claims 1, 19, and 27, Papadopoulos describes the use of conventional static web pages that may be updated with control system information and provided as snapshots to a user. In particular, Papadopoulos describes seven predefined web pages that may be presented to a user: "The home page contains hyperlinks to seven pages of data....The data appearing on the pages is static but can be automatically updated at preselected times." Papadopoulos, at Col. 8, l. 62 - Col. 9, l. 16.

There is no discussion in Papadopoulos of using non-markup languages to define a web page. Indeed, there is no discussion in Papadopoulos of non-markup languages or an interface that allows a user to design a Web site. The Office Action asserts that Figure 2, element 30, column 4, lines 9-12, teach the limitation of "storing said configuration data as a non-markup language Web site database." However, referring to the cited column and lines, Papadopoulos only describes that "[t]he web site 4 includes the network interface 16 having a unique Internet address 18 and a web server 30. The web server 30 provides the home page for the website." This does not support any discussion of the use of a non-markup language for defining a Web site. In contrast, it is consistent with the rest of the disclosure of Papadopoulos that describes the use of predefined web pages, i.e., conventional markup language web pages.

Applicants assert that Papadopoulos does not teach or describe a system, method, or apparatus for providing information regarding the operation of a control system that defines a web site using a non-markup language and storing that non-markup language as a non-markup language Web site database, as called for in Claims 1, 19, and 27. Additionally, applicants assert that Papadopoulos also does not teach or describe generating a web site using the stored non-markup language Web site database.

For the above reasons, applicants respectfully request withdrawal of the 35 U.S.C. § 102(e) rejection of Claims 1, 19, and 27.

(b) Claim 2

Claim 2 is dependent on Claim 1. As discussed above, Papadopoulos fails to teach or suggest each of the limitations recited in Claim 1. Accordingly, for the above-mentioned reasons, Claim 2 is likewise allowable over Papadopoulos.

Claim 2 also adds an additional limitation that further distinguishes it from the teachings of Papadopoulos. In particular, Claim 2, in a manner similar to Claim 27, includes the limitation "wherein the Web server module is operative to receive a request for a Web page of said Web site and to dynamically generate a markup language Web page from said non-markup language Web site database in response to a request."

As discussed above, Papadopoulos does not discuss dynamically generating web pages, nor does it discuss the use of a non-markup language to define a web site. In particular, Papadopoulos specifically describes that seven predefined web pages with static data may be provided to a user. In direct contrast, Claim 2 includes the limitation of dynamically generating a markup language Web page from the non-markup language Web site database.

Applicants assert that Papadopoulos, in addition to not teaching the limitations of Claim 1, also does not teach or describe a system that dynamically generates a markup language web page from said non-markup language Web site database in response to a request, as called for in Claim 2.

For the above reasons, applicants respectfully request withdrawal of the 35 U.S.C. § 102(e) rejection of Claim 2.

(c) Claims 3-18, 20-26, 28-31, and 33-37

Claims 3-18 each ultimately depend from Claim 1, Claims 28-31, Claims 20-26 ultimately depend from Claim 19, and Claims 33-37 each ultimately depend from Claim 27. As

discussed above, Papadopoulos fails to teach or suggest each of the limitations recited in Claims 1, 19, and 27. Accordingly, for the above-mentioned reasons, Claims 3-18, 20-26, 28-31, and 33-37 are likewise allowable over Papadopoulos. In addition, Claims 3-18, 20-26, 28-31, and 33-37 further add to the nonobviousness of the claims.

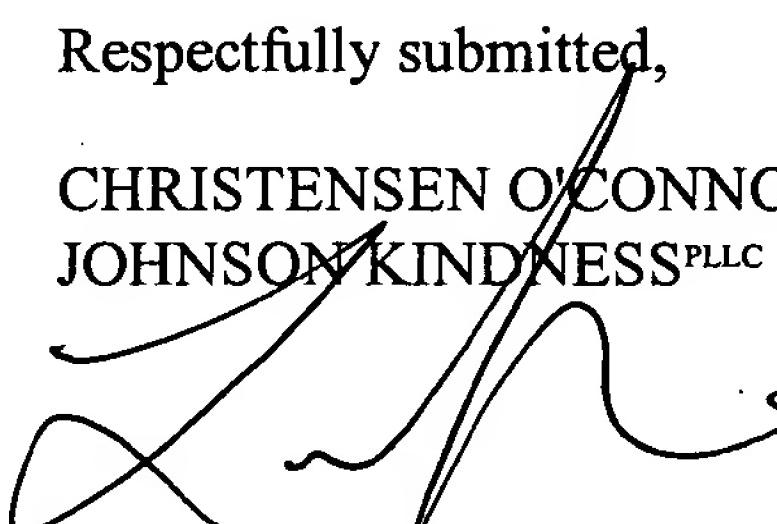
CONCLUSION

Based on the above-referenced arguments and amendments, applicants respectfully submit that all of the pending claims of the present application, Claims 1-31, and 33-42, are allowable over the cited and applied references. Because the cited and applied references fail to teach a method, system, and apparatus for providing information regarding the operation of a control system that utilizes non-markup language configuration data defining a Web site, storing the configuration data as a Web site database, and dynamically generating a web page from the non-markup language Web site database in response to a request, applicants respectfully request withdrawal of the rejections of the claims and allowance of the present application.

If any questions remain, applicants request that the Examiner contact the undersigned at the telephone number listed below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}

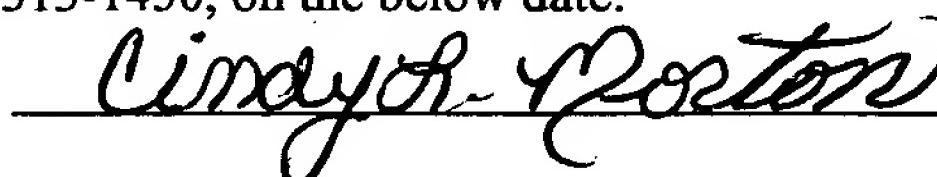


Larry T. Harris
Registration No. 44,745
Direct Dial No. 206.695.1642

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: December 30, 2004



LTH:clm

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue, Suite 2800
Seattle, Washington 98101
206.682.8100